CENTRE DE RENNES
IRISA

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél.:(3) 954 90 20

Rapports de Recherche

N° 230

ON EXACT AND APPROXIMATE ITERATIVE METHODS FOR GENERAL QUEUEING NETWORKS

Raymond A. MARIE
William J. STEWART

Juillet 1983

IRISA INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
Avenue du Général Leclerc
35042 - RENNES CÉDEX
FRANCE
Tél. : (99) 36.20.00
Télex : UNIRISA 95 0473 F

ON EXACT AND APPROXIMATE ITERATIVE METHODS FOR GENERAL QUEUEING NETWORKS

Raymond A. MARIE *
William J. STEWART **

Publication Interne n° 204
27 pages
Juin 1983

Résumé

La méthode itérative approchée de Marie fournit une approche viable pour résoudre des modèles de files d'attente d'une manière efficace et avec une précision généralement acceptable. D'un autre côté, la technique itérative de Takahashi, qui réalise une succession de phases d'aggrégation et de désaggrégation, calcule de façon exacte le vecteur des probabilités d'état stationnaires des chaînes de Markov. Cette dernière méthode n'est généralement pas adaptée à la résolution des réseaux de files d'attente visés ci-dessus. Dans ce papier, on montre comment la méthode de Takahashi peut être, en théorie, appliquée aux réseaux de files d'attente et, par là, on montre sa relation avec la méthode de Marie. Ceci fournit une autre présentation, rigoureuse du point de vue mathématique, de cette dernière méthode et permet de mieux caractériser les sources d'approximation.

Abstract

The approximate iterative method of Marie provides a viable approach for solving queueing network models efficiently and with reasonable accuracy. On the other hand, the iterative technique of Takahashi performs successive aggregation and disaggregation steps to compute the exact stationary probability vector of Markov chains but is generally not appropriate for solving queueing networks. In this paper, we show how the method of Takahashi may be applied to queueing networks and in doing so, demonstrate its relationship with the method of Marie, thereby enabling us to provide a more rigorous mathematical framework for the method of MARIE and to obtain some insight into the source of the approximation error.

* IRISA/INRIA Rennes
Campus de Beaulieu - 35042, RENNES CEDEX, FRANCE
**Dept of Computer Science - N. Carolina State University - Raleigh, NC 27650 - U.S.A.

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE (L.A. 227)
UNIVERSITÉ DE RENNES 1 I.N.S.A. DE RENNES
INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE (LABORATOIRE DE RENNES)

On Exact and Approximate Iterative Methods

for General Queueing Networks

Raymond A. Marie,
William J. Stewart.

1. Introduction

Queueing network models have proven themselves to be valuable tools in the analysis of complex systems. For example, their application to the performance evaluation of computer and communication systems is well documented in the literature. However, despite the success achieved by these queueing network models, their applicability is limited by the restrictions which must be imposed on the model to render it tractable. Perhaps one of the most incidious restrictions is that stations of the network which serve customers according to a first-come first-served scheduling discipline must have a service time distribution that is exponentially distributed. Currently, the most general networks for which we can obtain analytic solutions are the so called product form [BASK75]. The solution to these networks may be written as a product of terms, each term corresponding to the solution of an individual station of the network, and in such cases the global solution may be found very efficiently. These networks are said to be separable. Since the class of network which satisfy product form is very restrictive, a different approach to solving nonseparable networks must be found.

One approach which has received much attention over the past decade is to use an approximation technique whereby a solution which is (hopefully) close to the exact solution of the queueing network is determined. Among these methods are the iterative heuristics proposed by Chandy, Herzog and Woo [CHAN75] and by Marie [MARI78]. Substantial experimental testing of these methods over the past 5-10 years indicate that the accuracy achieved by these methods is commensurate with that of the modelling process itself. In this paper we shall largely be concerned with the iterative approach proposed by Marie. In his thesis, [BALB79], Balbo shows this to be the method of choice.

The approximation approach is not the only possibility. It is always possible, at least from a theoretical point of view, to formulate the queueing network as a Markov process and to use numerical techniques to obtain its stationary probability vector, [STEW78]. From this vector, all the required measures of effectiveness of the network may be derived. In most cases, however, the dimensionality of the problem is so great that in practice such a solution becomes infeasible. It then becomes necessary to decompose the problem into smaller problems, to determine the solution to each of the smaller problems and then to combine these solutions into a global solution, i.e. the "divide and conquer" principle. This approach, which yields only an approximate result, has been developed by Courtois [COUR77] and by Stewart [STEW80], but unlike the iterative methods of Chandy, Herzog, Woo and Marie, error bounds have been derived. More recent research has been directed at incorporating these approaches into numerical iterative procedures, such as block Gauss-Seidel or the power method [KOUR83, MCAL83] so that they are made to converge onto the exact solution. Somewhat different iterative approaches have been developed by Takahashi [TAKA75] and Vantilborgh [VANT81].

The iterative method of Takahashi shall play a central role in the current paper. Each iteration of this method consists of both an aggregation step and a disaggregation step. The purpose of the aggregation step is to determine the relative probabilities of being in different subsets of states of the underlying Markov chain, while the disaggregation step is an attempt to distribute the probability of being in a given subset among the states of that subset. We shall consider this method in more detail in the next section. Also in the next section we shall introduce a reference example which we shall use throughout the paper for the purpose of illustration.

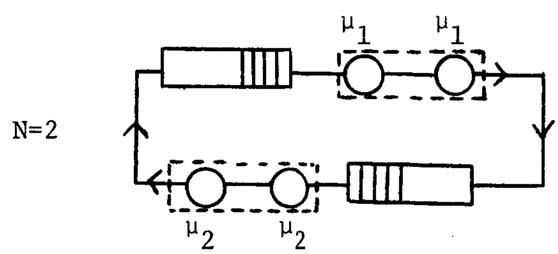
In section 3, we shall describe the queueing network model and very briefly outline, in algorithmic form, the approximate iterative method of Marie. We then consider the application of the exact iterative method of Takahashi to the queueing network. We shall see that, for this specific Markov process, the aggregation matrix consists of only three different types of element, and this entails some simplification of the method. We should recall at this point that a numerical method such as that of Takahashi, which works directly on the state space of the underlying Markov chain, is not an appropriate method for most queueing networks, due to the number of states generated.

The main result of this paper is to exhibit the relationship between the methods of Marie and of Takahashi and this is achieved in the final part of section 3. By putting the heuristic approach of Marie on a more rigorous framework and by providing some insight into the source of the approximation error, it is hoped that this work will pave the way for further research to yield error bounds in an initial stage and perhaps at some future date, the derivation of exact iterative methods which do not need to work on the underlying Markov state space.

2. The Method of Takahashi

2.1. The Reference Example

Throughout this paper we shall illustrate the two approaches by means of a simple two station closed queueing network which we shall refer to as the reference example. Each station consists of a single Erlang-2 server while two customers of the same class circulate in the model. The service rates are as indicated in the figure below and we shall define $\phi \triangleq \frac{\mu_1}{\mu_2}$.



A state of the system may be described by means of the triplet (n,ℓ_1,ℓ_2) where $n \in \{0,1,2\}$ denotes the number of customers at station 1 and $\ell_1 \in \{1,2\}$ (respectively $\ell_2 \in \{1,2\}$) denotes the phase of service of the customer in service at station 1, (respectively station 2). It is easy to verify that this representation generates a total of 8 states and that the associated infinitesimal generator matrix T is given by

$$
\begin{array}{c|cc|cccc|cc|}
\underline{\text{State}} & (001) & (002) & (111) & (112) & (121) & (122) & (210) & (220) \\
\hline
(001) & \frac{1}{\phi+1} & \frac{1}{\phi+1} & & & & & & \\
(002) & & \frac{1}{\phi+1} & \frac{1}{\phi+1} & & & & & \\
\hline
(111) & & & -1 & \frac{1}{\phi+1} & \frac{\phi}{\phi+1} & & & \\
(112) & & & & -1 & & \frac{\phi}{\phi+1} & \frac{1}{\phi+1} & \\
(121) & \frac{\phi}{\phi+1} & & & & -1 & \frac{1}{\phi+1} & & \\
(122) & & \frac{\phi}{\phi+1} & & & & -1 & & \frac{1}{\phi+1} \\
\hline
(210) & & & & & & & \frac{1}{\phi+1} & \frac{1}{\phi+1} \\
(220) & & & \frac{\phi}{\phi+1} & & & & & \frac{\phi}{\phi+1} \\
\hline
\end{array}
$$

The associated stationary probability distribution vector p, is obtained by solving $p^tT=0$; $p^te=1$. Here e is a vector all of whose elements are 1, and p^t denotes the transpose of the vector p. It may be verified that the vector $(\phi^3(\phi+1),\ \phi^3(\phi+3),\ \phi(\phi+1)^2,\ \phi(\phi+1),\ \phi^2(\phi+1),\ 2\phi^2,\ \phi+1,\ 3\phi+1)^t$ satisfies $p^tT=0$ but not $p^te=1$.

2.2. The Iterative Method

The method of Takahashi [TAKA75] is an iterative method in which each iteration consists of an aggregation step, a disaggregation step and a test for convergence. The method entails a partitioning of the state space into blocks of states. In the example we shall partition the state space as indicated by the dashed lines in the infinitesimal generator matrix T. This partitioning remains fixed during the execution of the algorithm. We shall use z to denote the number of states; r to denote the number of blocks and z_i to denote the number of states in block i. Thus $\sum_{i=1}^{r} z_i = z$. In the example we have $z=8$, $r=3$, $z_1=2$, $z_2=4$ and $z_3=2$. We shall let e_i denote the vector of length z_i, all of whose elements are unity. Consider

$$(p_1^t, p_2^t, \ldots, p_r^t) \begin{bmatrix} T_{11} & T_{12} & \cdots & T_{1r} \\ T_{21} & T_{22} & \cdots & T_{2r} \\ \vdots & \vdots & & \vdots \\ T_{r1} & T_{r2} & \cdots & T_{rr} \end{bmatrix} = 0$$

where $p^t = (p_1^t, p_2^t, \ldots, p_r^t) \in \mathbb{R}^z$ is the stationary probability vector, with $p_i \in \mathbb{R}^{z_i}$ and $T_{ij} \in \mathbb{R}^{z_i x z_j}$ is the i-jth block of the partitioned infinitesimal generator matrix T. We shall denote by π_i the i-th element of the vector p; i.e. $p^t = (p_1^t, p_2^t, \ldots, p_r^t) = (\pi_1, \pi_2, \ldots, \pi_z)$

Let us define $w^t = (\omega_1, \omega_2, \ldots \omega_r) \in \mathbb{R}^r$, where $\omega_i \stackrel{\Delta}{=} p_i^t e_i$. Then ω_i is the probability of being in one of the states of block i; i.e.

$\omega_i = \sum_{j=\sigma}^{\sigma+z_i} \pi_j$ and $\sigma = \sum_{j=1}^{i-1} z_j$. Furthermore, if $b^t \stackrel{\Delta}{=} (\frac{1}{\omega_1} p_1^t, \frac{1}{\omega_2} p_2^t, \ldots \frac{1}{\omega_r} p_r^t) \stackrel{\Delta}{=} (b_1^t, b_2^t, \ldots b_r^t)$, then the jth element of b_i is the conditional probability of being in state j of block i at steady state given that the Markov process is in block i.

In terms of the example we find

$$w^t = (\omega_1, \omega_2, \omega_3) = (\pi_1+\pi_2, \pi_3+\pi_4+\pi_5+\pi_6, \pi_7+\pi_8)$$
$$\sim (2\phi^3(\phi+2),\ 2\phi[(\phi+1)^2+\phi],\ 2(\phi+2)),\ \text{and}$$

$$b^t = \left(\frac{\pi_1}{\pi_1+\pi_2}, \frac{\pi_2}{\pi_1+\pi_2} \;\Big|\; \frac{\pi_3}{\pi_3+\pi_4+\pi_5+\pi_6}, \frac{\pi_4}{\pi_3+\pi_4+\pi_5+\pi_6}, \frac{\pi_5}{\pi_3+\pi_4+\pi_5+\pi_6}, \frac{\pi_6}{\pi_3+\pi_4+\pi_5+\pi_6} \;\Big|\; \frac{\pi_7}{\pi_7+\pi_8}, \frac{\pi_8}{\pi_7+\pi_8}\right)$$

$$= \left(\frac{\phi+1}{2(\phi+2)}, \frac{\phi+3}{2(\phi+2)} \;\Big|\; \frac{(\phi+1)^2}{2[(\phi+1)^2+\phi]}, \frac{\phi+1}{2[(\phi+1)^2+\phi]}, \frac{\phi(\phi+1)}{2[(\phi+1)^2+\phi]}, \frac{\phi}{(\phi+1)^2+\phi} \;\Big|\; \frac{\phi+1}{2(\phi+2)}, \frac{3\phi+1}{2(\phi+2)}\right) \quad (1)$$

The purpose of the aggregation step is to determine an approximation to the vector w; in other words to obtain an approximation to the stationary probabilities of being in each of the blocks. This is achieved by constructing a matrix which represents the transition rate between the various blocks and determining the stationary probability vector of this matrix. This matrix Q is defined as follows:

$Q \triangleq UTV \in \mathbb{R}^{r \times r}$ where $U \in \mathbb{R}^{r \times z}$, $V \in \mathbb{R}^{z \times r}$ and

$$U = \begin{pmatrix} b_1^t & & & 0 \\ & b_2^t & & \\ & & \ddots & \\ 0 & & & b_r^t \end{pmatrix}, \qquad V = \begin{pmatrix} e_1 & & & 0 \\ & e_2 & & \\ & & \ddots & \\ 0 & & & e_r \end{pmatrix}$$

Note that the i-jth element of Q is

$$(Q)_{ij} = b_i^t T_{ij} e_j$$

and that $w^t Q = 0$

$$(\text{since } w^t Q = w^t UTV = p^t TV = 0).$$

Unfortunately, it is not possible to construct Q exactly without knowing the stationary probability vector p. However, each iteration will yield successively better approximations to p and therefore to Q.

In terms of the example,

$$\begin{bmatrix} \frac{\pi_1}{\pi_1+\pi_2}, & \frac{\pi_2}{\pi_1+\pi_2}, & 0, & 0, & 0, & 0, & 0, & 0 \\ 0, & 0, & \frac{\pi_3}{\pi_3+\pi_4+\pi_5+\pi_6}, & \frac{\pi_4}{\pi_3+\pi_4+\pi_5+\pi_6}, & \frac{\pi_5}{\pi_3+\pi_4+\pi_5+\pi_6}, & \frac{\pi_6}{\pi_3+\pi_4+\pi_5+\pi_6}, & 0, & 0 \\ 0, & 0, & 0, & 0, & 0, & 0, & \frac{\pi_7}{\pi_7+\pi_8}, & \frac{\pi_8}{\pi_7+\pi_8} \end{bmatrix}$$

$$TV = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{1}{\phi+1}, & \frac{1}{\phi+1}, & 0 \\ 0, & 0, & 0 \\ 0, & -\frac{1}{\phi+1}, & \frac{1}{\phi+1} \\ \frac{\phi}{\phi+1}, & -\frac{\phi}{\phi+1}, & 0 \\ \frac{\phi}{\phi+1}, & -1, & \frac{1}{\phi+1} \\ 0, & 0, & 0 \\ 0, & \frac{\phi}{\phi+1}, & \frac{\phi}{\phi+1} \end{bmatrix}, \text{ and}$$

$$Q = \begin{bmatrix} \frac{\pi_2}{\pi_1+\pi_2}(\frac{1}{\phi+1}) & , \frac{\pi_2}{\pi_1+\pi_2}(\frac{1}{\phi+1}) & , 0 \\ \frac{\pi_5+\pi_6}{\pi_3+\pi_4+\pi_5+\pi_6}(\frac{\phi}{\phi+1}) & , \frac{-1}{\pi_3+\pi_4+\pi_5+\pi_6}(\frac{\pi_4}{\phi+1} + \frac{\phi\pi_5}{\phi+1} + \pi_6) & , \frac{\pi_4+\pi_6}{\pi_3+\pi_4+\pi_5+\pi_6}(\frac{1}{\phi+1}) \\ 0 & , \frac{\pi_8}{\pi_7+\pi_8}(\frac{\phi}{\phi+1}) & , -\frac{\pi_8}{\pi_7+\pi_8}(\frac{\phi}{\phi+1}) \end{bmatrix}$$

Note that the elements of Q give the rate of transitions between blocks.

This completes the aggregation aspect of the method. If the vectors b_1, b_2 and b_3 are known, the matrix Q can be determined and hence the vector w can be computed.

The purpose of the disaggregation step is to determine the conditional stationary probabilities b_k of the states of block k. These probabilities are determined under the assumption that the conditional probabilities are known for all the other blocks (i.e. given b_ℓ $\ell \neq k$) and when the probability distribution among the blocks themselves (i.e. w) is also known. This process must be repeated for all blocks k = 1,2,...r. Since the vectors b_ℓ, $\ell \neq k$ are not known a priori, they must be estimated and on successive iterations, will tend to converge to their true values.

Takahashi shows that b_k may be determined from the relationship

$$(p_k^t,\ 1-w_k)Q_k = 0.$$

Q_k may be thought of as an infinitesimal generator and is defined by

$$Q_k = \begin{bmatrix} T_{kk} & h_k \\ g_k^t & \gamma_k \end{bmatrix} \quad \text{where } h_k = -T_{kk}e_k$$

$$g_k = \frac{1}{1-w_k} \sum_{\ell \neq k} w_\ell b_\ell^t T_{\ell k}$$

$$\text{and } \gamma_k = -g_k^t e_k$$

Note that the vector h_k is chosen so that the sum of the elements in the first z_k rows of Q_k is zero. The i-th element of h_k gives the rate at which the process exits block k from the i-th state of this block. The i-th element of the vector g_k gives the rate at which the process enters state i of block k from outside this block while the element γ_k is chosen so that the sum of the elements in the last row is zero.

Let us consider the first block of the example. Then since

$$\frac{1}{1-\omega_1}[\omega_2 b_2^t T_{21} + \omega_3 b_3^t T_{31}] = \frac{1}{1-\omega_1}[p_2^t T_{21} + p_3^t T_{31}]$$

$$= \frac{1}{1-(\pi_1+\pi_2)}[(\pi_5(\frac{\phi}{\phi+1}), \pi_6(\frac{\phi}{\phi+1})) + (0,0)]$$

we obtain

$$Q_1 = \begin{bmatrix} -\frac{1}{\phi+1} & , & \frac{1}{\phi+1} & , & 0 \\ 0 & , & -\frac{1}{\phi+1} & , & \frac{1}{\phi+1} \\ \frac{\pi_5}{1-(\pi_1+\pi_2)}(\frac{\phi}{\phi+1}) & , & \frac{\pi_6}{1-(\pi_1+\pi_2)}(\frac{\phi}{\phi+1}) & , & -\frac{(\pi_5+\pi_6)}{1-(\pi_1+\pi_2)}(\frac{\phi}{\phi+1}) \end{bmatrix}$$

Note that $(p_1^t, 1-\omega_1)Q_1 = 0$

since

$$(p_1^t, 1-\omega_1)Q = (1-\omega_1)(\frac{\pi_1}{1-\omega_1}, \frac{\pi_2}{1-\omega_1}, 1)Q$$

$$= (1-\omega_1)\left[\frac{\phi\pi_5-\pi_1}{(1-\omega_1)(\phi+1)}, \frac{\pi_1+\phi\pi_6-\pi_2}{(1-\omega_1)(\phi+1)}, \frac{\pi_2-\phi(\pi_5+\pi_6)}{(1-\omega_1)(\phi+1)}\right]$$

$$= \frac{1}{\phi+1}[\phi\pi_5-\pi_1, \pi_1+\phi\pi_6-\pi_2, \pi_2-\phi(\pi_5+\pi_6)]$$

$$= \frac{1}{\phi+1}[0, 0, 0]$$

Thus we can form Q_k (since we know w and b_ℓ $(\ell \neq k)$) and from it we can compute $(p_k^t, 1-\omega_k)$. This gives us p_k and we can find b_k since

$$b_k = \frac{1}{\omega_k} p_k$$

When b_k has been found for k = 1,2,...r, the disaggregation aspect of the algorithm is considered to be finished.

To conclude this section, we present the iterative method of Takahashi in the following algorithmic form:

Given initial approximations $b_k^{(0)}$ $k = 1,2,\ldots r$

(Iteration n)

(1) Aggregation Step:

(i) Form $Q^{(n)} = U^{(n)}TV$ i.e. $(Q^{(n)})_{ij} = b_i^{t^{(n-1)}} T_{ij} e_j$

(ii) Solve $w^{(n)}Q^{(n)} = 0$ with $w^{(n)}e = 1$

(2) Disaggregation Step:

For $k = 1,2,\ldots r$, calculate $b_k^{(n)}$; i.e.

(i) Form $Q_k^{(n)}$ - use $b_\ell^{(n)}$ $\ell \leq k-1$

and $b_\ell^{(n-1)}$ $\ell \geq k-1$

(ii) Get $(p_k^t, 1-\omega_k)$ from $(p_k^t, 1-\omega_k)Q_k^{(n)} = 0$

(iii) Form $b_k^{(n)} = \frac{1}{\omega_k} p_k$

(3) Test for Convergence:

-if satisfied, STOP

if not, goto (1)

3. Iterative Methods for Queueing Networks

3.1. Description of the Queueing Network

Consider a closed queueing network R consisting of M stations in which N customers circulate according to the probabilities of a fixed routing matrix. Each station i, $i = 1,2,\ldots,M$, contains a single server who provides service according to a FCFS scheduling discipline. We shall assume that the service time distribution at station i may be represented by a law of Cox of order L_i. This law may be characterized by two vectors of length L_i:

$(\mu_{i1},\mu_{i2},\ldots,\mu_{iL_i})$ and $(a_{i1},a_{i2},\ldots,a_{iL_i})$. The parameter μ_{ij} denotes the rate of transition from the j-th service phase and $(1-a_{ij})$ is the probability that the service will finish after the j-th phase has been completed. Note that $a_{iL_i} \equiv 0$, $\forall i$, and that phase 1 is always completed.

For this class of queueing network, there exists a Markov process which allows us to characterize the state of the network and its evolution in time. Any state of this Markov process may be described by a vector of length 2M as follows:

$$s = (n_1,\ell_1,n_2,\ell_2,\ldots,n_M,\ell_M),$$

where n_i represents the number of customers at station i and ℓ_i is its current phase of service.

We shall denote by S the set of all possible states s, and since the network is closed, it follows that

$$\sum_{i=1}^{M} n_i = N, \ \forall\, s \in S .$$

The variable ℓ_i assumes its values from the set $\{0,1,\ldots,L_i\}$ if, by convention, we allow $\ell_i=0$ to signify that server i is idle. Note that $\ell_i=0$ iff $n_i=0$.

As in the previous section, z will denote the number of states of the Markov process; i.e. $z = |S|$, the cardinality of the set S. We shall denote by T, the infinitesimal generator matrix of the Markov process and p its stationary vector. It follows, therefore, that $p^tT=0$. Needless to say, we shall assume that all conditions necessary for p to exist are satisfied; (irreducible routing matrix, finite mean service time).

3.2. The Approximate Iterative Method

For the class of queueing network described in section 3.1, the approximate iterative method presented by Marie [MARI78] has the following algorithmic form.

Given an initial vector of load dependent service rates;

$$\nu^{(0)} = (\nu_1^{(0)}(1),\ \nu_1^{(0)}(2),\ \ldots,\ \nu_1^{(0)}(N),\ \nu_2^{(0)}(1),\ \ldots,\ \nu_M^{(0)}(N))^t$$

where $\nu_i^{(0)}(j)$ indicates the service rate at station i when there are j customers at that station, then

At iteration k

(1) Determine a set of load dependent arrival rates.

This is achieved by an analysis of the product form queueing network which has the same topology as R and for which the service rates are given by the vector $\nu^{(k-1)}$. The vector of load dependent arrival rates $\lambda^{(k)} = (\lambda_1^{(k)}(1),\ \lambda_1^{(k)}(2),\ldots,\ \lambda_1^{(k)}(N),\ \lambda_2^{(k)}(1),\ldots,\ \lambda_M^{(k)}(N))^t$ may be computed by means of the usual algorithms for closed product form queueing networks.

(2) Determination of marginal probabilities

(i.e. calculation of the throughputs).

For i = 1,2,...,M, we can find the vector of throughputs

$$(\nu_i^{(k)}(1),\ \nu_i^{(k)}(2),\ \ldots,\ \nu_i^{(k)}(N))$$

from an analysis of the queue $\lambda_i^{(k)}(j)|C_{L_i}|1|N$. Note that if further iterations are required, these will become the load dependent service rates at station i in the associated product form network.

(3) Test for convergence:

- if satisfied, STOP;

otherwise, initiate iteration (k+1)

3.3. An Exact Iterative Method

Let us consider a partition on the set S such that $S = \bigcup S_1(j,k)$ where

$$S_1(j,k) = \{s|n_1=j \text{ and } \ell_1=k\}.$$

The subscript 1 is used to denote that station 1 has been tagged for consideration. Any other station would have been equally appropriate. We shall assume that the states of S are arranged in the following partial ordering:

- states of $S_1(j,k-1)$ preceed those of $S_1(j,k)$ $\forall$ k = 1,2,...L_1.
- states of $S_1(j-1,k)$ preceed those of $S_1(j,k)$ $\forall$ j = 1,2,...N.

Consequently, all the states of a given subset $S_1(j,k)$ are consecutive. For our purposes any total ordering on the set S which possesses this partial ordering is satisfactory. Having chosen such a total ordering, we shall denote by s_j the j-th state of S accordingly.

To be consistent with the notation used in the previous section, we shall call each subset a block. Block i denotes the i-th subset $S_1(j,k)$ chosen according to the above ordering. By construction, the number of blocks, r, is equal to (NL_1+1). We shall denote $n_1(i)$, (respectively $\ell_1(i)$) the value of n_1 (respectively ℓ_1) for the states of block i. In the suite, in order to keep the notation reasonably simple, we shall omit the subscript 1 which is the explicit reference to station 1. We shall include it only when it becomes necessary. Thus $n_1(i) \equiv n(i)$ and $\ell_1(i) \equiv \ell(i)$.

As in section 2, z_i shall denote the number of states in block i. Therefore, we have

$$z_i = |S(n(i),\ell(i))| \ , \quad i = 1,2,\ldots r$$

Finally, we shall denote by B(i) the set of order indices of the states

$$s \in S(n(i),\ell(i))$$

Returning to the reference example, the partition chosen corresponds to

$$S = S(0,0) \cup S(1,1) \cup S(1,2) \cup S(2,1) \cup S(2,2).$$

with $S(0,0) = \{(0,0,2,1),\ (0,0,2,2)\}$,

$S(1,1) = \{(1,1,1,1),\ (1,1,1,2)\}$,

$S(1,2) = \{(1,2,1,1),\ (1,2,1,2)\}$,

$S(2,1) = \{(2,1,0,0)\}$, and

$S(2,2) = \{(2,2,0,0)\}$

We say that the Markov process is in block 3 if the process is in one of the states of the subset S(1,2). In this case we have n(3)=1 and $\ell(3)=2$. Finally, if we keep the order on S that was used in section 2 (for this order is compatible with the partial ordering defined in the current section), we have B(3)={5,6}.

We shall now compute the probabilities ω_i, i=1,2,...,r that the process at steady state is in one of the states of block i by means of the aggregation step suggested by Takahashi. We shall consider the matrix $V \in \mathbb{R}^{z\times r}$ whose elements are given by

$$(V)_{ij} = \begin{cases} 1 & \text{if } i \in B(j) \\ 0 & \text{otherwise.} \end{cases}$$

and the matrix $U \in \mathbb{R}^{rxz}$ such that

$(U)_{ij}$ = conditional probability of being in state s_j knowing that the process is in block i.

i.e. $$(U)_{ij} = \begin{cases} \dfrac{\pi_j}{\sum\limits_{k \in B(i)} \pi_k} & \text{if } j \in B(i) \\ 0 & \text{otherwise} \end{cases}$$

For the reference example, we have

$$V = \begin{bmatrix} 1 & & & \\ 1 & & & \\ & 1 & & \\ & 1 & & \\ & & 1 & \\ & & 1 & \\ & & & 1 \\ & & & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{pmatrix} ** & & & & & & & \\ & & ** & & & & & \\ & & & & ** & & & \\ & & & & & & * & \\ & & & & & & & * \end{pmatrix}$$

where only elements which are non zero are indicated. The matrix corresponding to the matrix product TV is of dimension (zxr) and has coefficients given by:

$(TV)_{ij}$ = transition rate from state s_i to block j

$= \sum\limits_{k \in B(j)} t_{ik}$, for $i \neq j$ and t_{ik} is the i-kth element of T.

$(TV)_{ii} = - \sum\limits_{j \neq i} (TV)_{ij}$ otherwise.

The matrix Q which corresponds to the product UTV is a square matrix of order r and has coefficients

$i \neq j$: $(Q)_{ij} = \sum\limits_{k \in B(i)}$ (probability of being in state s_k | process is in block i)

*(transition rate from state s_k to block j)

$$= \Big(\sum_{k \in B(i)} \pi_k \Big(\sum_{\ell \in B(j)} t_{k\ell}\Big)\Big) \Big/ \sum_{k \in B(i)} \pi_k$$

= Transition rate from block i to block j (knowing that the process is in block i)

$i=j$: $\quad (Q)_{ii} = -\sum_{j \neq i} (Q)_{ij}$

It is now necessary to take the specific Markov process considered here into account and the effect of this specific process on the elements $(Q)_{ij}$. For any block i, the only transitions which are possible from block i to block j, $j \neq i$, belong to one of the three following types:

type 1: if $n(j) = n(i)+1$ and $\ell(j) = \ell(i)$

type 2: if $n(j) = n(i)$ and $\ell(j) = \ell(i)+1$

type 3: if $n(j) = n(i)-1$ and $\ell(j) = 1$ or 0.

The possibility $\ell(j)=0$ corresponds to the case where $n(j)=0$.
We may therefore refer to interblock transitions of type 1, 2 or 3. Note that, as a result, there will only be three non zero off-diagonal elements per row, which implies that for large r, Q will be very sparse.

We shall now show that interblock transition rates of type 2 and 3 are independent of the stationary probability vector p.
For a transition of type 2 between blocks i and j, we have

$$\forall k \varepsilon B(i), \sum_{\ell \varepsilon B(j)} t_{k\ell} = \mu_{\ell(i)} a_{\ell(i)} \text{ , where } \qquad \mu_{\ell(i)} \equiv \mu_{1\ell_1(i)}$$

$$\text{and } \quad a_{\ell(i)} \equiv a_{1\ell_1(i)}$$

In other words, a transition from any state of block i to block j which results from a change of phase of the server in station 1 has a constant rate. It follows that

$$(Q)_{ij} = \mu_{\ell(i)} a_{\ell(i)}$$

Note that in this case, $j=i+1$.

This rate is therefore independent of the probabilities π_j, and depends only on the defining characteristics of the network.

Likewise, for a transition of type 3, which corresponds to a service completion for server 1, we have

$$\forall k \varepsilon B(i), \sum_{\ell \varepsilon B(j)} t_{k\ell} = \mu_{\ell(i)}(1-a_{\ell(i)})$$

and in this case we obtain

$$(Q)_{ij} = \mu_{\ell(i)}(1-a_{\ell(i)})$$

which is again independent of the probabilities π_j.

Consequently, taking into account the specific application in which we are interested, only the computation of transition rates of type 1 require the vector p (or an approximation to p in the case of the iterative approach of Takahashi).

Transitions of type 1 correspond to the arrival of a customer in station 1. We shall now introduce a specific notation for this type of interblock transition:

Let $\lambda(u,v) \triangleq (Q)_{ij}$

for the case when $n(i)=u$; $n(j)=u+1$

and $\ell(i)=v=\ell(j)$.

For the reference example, the interblock transition rate matrix Q may be written

$(n(i),\ell(i))$	block	1	2	3	4	5
(0,0)	1	$-\lambda(0,0)$	$\lambda(0,0)$	0	0	0
(1,1)	2	0	$-(\mu_1+\lambda(1,1))$	μ_1	$\lambda(1,1)$	0
(1,2)	3	μ_1	0	$-(\mu_1+\lambda(1,2))$	0	$\lambda(1,2)$
(2,1)	4	0	0	0	$-\mu_1$	μ_1
(2,2)	5	0	μ_1	0	0	$-\mu_1$

where, by definition

$$\lambda(0,0) = \frac{\mu_2\pi_2}{\pi_1+\pi_2} \quad ; \quad \lambda(1,1) = \frac{\mu_2\pi_4}{\pi_3+\pi_4} \quad ; \quad \lambda(1,2) = \frac{\mu_2\pi_6}{\pi_5+\pi_6} \qquad (2)$$

With application to the type of network considered, the aggregation step of Takahashi's method therefore consists of determining the vector w which is the solution of

$$w^tQ = 0$$

where the only parameters which are, a priori, unknown, (but which may be expressed in terms of the vector p) are the rates

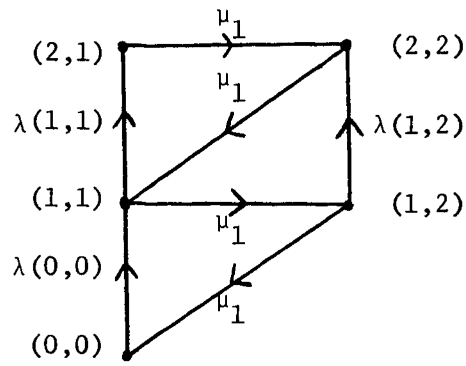
$$\lambda(n,\ell),\ n=0,1,\ldots,N-1; \quad \ell=0,1,\ldots,L_1.$$

It is interesting to note that the Markovian process associated with the blocks and with the vector w is identical to the Markov process associated with the states of the queue $\lambda(n,\ell)|C_{L_1}|1|N$. For this queue, the service time distribution is the same as that of station 1, and the arrival process is a stochastic point process for which the intensity depends on the state (n,ℓ) of the queue.

Rather than use the standard numerical technique to obtain the vector w (as in the case with Takahashi), we may use a method which is more suitable for this type of process, for example the recursive methods presented by [MARI83]. If $x(n,\ell)$ denotes the stationary probability that the queue is in state (n,ℓ), then the recurrence relation is obtained by using the fact that for any queue $\lambda(n,\ell)|C_L|1|N$ we have [MARI78]

$$\sum_{\ell=1}^{L} x(n+1,\ell)\mu_\ell(1-a_\ell) = \sum_{\ell=1}^{L} x(n,\ell)\lambda(n,\ell)$$

Such an approach is efficient both in terms of execution time and computer storage. For the reference example, instead of solving the system $w^tQ=0$ (which in this case is easy to obtain given the small number of variables ω_i), we may consider the Markov process associated with the queue $\lambda(n,\ell)|C_{L_1}|1|2$. The transition graph in this case, is as follows:



Naturally, the Chapman-Kolmogorov equations obtained by considering the queue are identical to those which would be obtained in expanding the system $w^tQ=0$, but here it is easier to generate the equations and their solution by the recurrent approach is readily obtained. The relation which allows us to use a recurrent method may be written here as

$$\mu_1 x(2,2) = \lambda(1,1)\ x(1,1) + \lambda(1,2)\ x(1,2)$$

$$\text{or} \quad \mu_1\omega_5 = \lambda(1,1)\ \omega_2 + \lambda(1,2)\ \omega_3 \tag{3}$$

(recall that $\omega_1 \equiv x(0,0)$; $\omega_2 \equiv x(1,1)$; $\omega_3 \equiv x(1,2)$; $\omega_4 \equiv x(2,1)$ and $\omega_5 \equiv x(2,2)$)

The first two Chapman-Kolmogorov equations, taken in the order of the blocks, may be written

(a) $\lambda(0,0)\omega_1 = \mu_1\omega_3$

(b) $(\lambda(1,1)+\mu_1)\omega_2 = \lambda(0,0)\omega_1 + \mu_1\omega_5$

Using equation (3) and (a), the second equation (b) may be written as

$$(\lambda(1,1) + \mu_1)\ \omega_2 = \lambda(0,0)\ \omega_1 + \lambda(1,1)\ \omega_2 + \lambda(1,2)\ \frac{\lambda(0,0)}{\mu_1}\ \omega_1$$

$$\text{i.e.} \qquad \omega_2 = \omega_1\ \frac{\lambda(0,0)}{\mu_1}\ [1 + \frac{\lambda(1,2)}{\mu_1}]$$

Also, from (a), we have directly

$$\omega_3 = \omega_1 \frac{\lambda(0,0)}{\mu_1}$$

From the remaining Chapmann-Kolmogorov equations, we obtain immediately

$$\omega_4 = \omega_2 \frac{\lambda(1,1)}{\mu_1}$$

$$\omega_5 = \omega_4 + \omega_3 \frac{\lambda(1,2)}{\mu_1}$$

For this particular example, knowledge of a vector proportional to p (see section 2) allows us to determine exactly the rates of transition of type 1. From equation (1) of the previous section we have

$$\lambda(0,0) = \frac{\mu_2(\phi+3)}{2(\phi+2)} \; ; \; \lambda(1,1) = \frac{\mu_2}{(\phi+2)} \; ; \; \lambda(1,2) = \frac{2\mu_2}{(\phi+3)}$$

It is therefore possible to verify that the solution obtained is indeed the same as that obtained from the vector proportional to p. For example, from (1) we have

$$\frac{\omega_3}{\omega_1} = \frac{\pi_3+\pi_4}{\pi_1+\pi_2} = \frac{\phi^2(\phi+3)}{2\phi^3(\phi+2)} = \frac{(\phi+3)}{2\phi(\phi+2)}$$

while from the results just calculated we find

$$\frac{\omega_3}{\omega_1} = \frac{\lambda(0,0)}{\mu_1} = \frac{\mu_2}{\mu_1} * \frac{(\phi+3)}{2(\phi+2)} = \frac{(\phi+3)}{2\phi(\phi+2)}$$

Unfortunately, in the general case, the vector p will not be known, so that, applying the method of Takahashi, it is necessary to follow this aggregation step with r disaggregation steps, (according to the procedure presented in section 2), in order to be able to calculate the r vectors p_i.

In this paper we shall not take up the development of the disaggregation steps. Their implementation in the case of network studied here does not appear to generate any significant simplification. It should, nevertheless, be noted that the vector g_k, belonging to the last row of the matrix Q_k depends only on the probabilities π_j of states contained in the neighboring blocks of block k. Thus, from sparsity considerations, this permits us to simplify the calculation of g_k.

What is important to note is that to use the method of Takahashi, it is necessary to have at least an approximation to the vector p. This implies that the method can only be used if the number of states in S is not excessive (e.g. 10^4, 10^5). However, for the type of network which we are concerned with here, it is not unusual to generate 10^{10}, 10^{100} states. The interest in developing approximate methods for solving such networks which do not need the vector p, should therefore be evident.

3.4. The relationship between the exact and approximate methods

Which might appear to be a reasonable first approximation is to replace each rate $\lambda(j,k)$ by $\lambda(j)$ where $\lambda(j)$ is defined to be the rate of transition from the block formed by those states s for which $n_1=j$, towards the block formed from states s for which $n_1=j+1$. In the reference example, it is possible to calculate $\lambda(0)$ and $\lambda(1)$ exactly. The blocks to be considered here are precisely those presented in section 2. We obtain $\lambda(0) = \lambda(0,0) = \dfrac{\mu_2(\phi+3)}{2(\phi+2)}$

$$\lambda(1) = \frac{\mu_2\pi_4+\mu_2\pi_6}{\pi_3+\pi_4+\pi_5+\pi_6} = \mu_2\frac{(3\phi+1)}{2[(\phi+1)+\phi]}$$

However, if we use these rates to generate a matrix $\hat{Q}$, then the solution $\hat{w}$ obtained from $\hat{w}^t\hat{Q}=0$ is only an approximation to the vector w. We have thus not made any progress, for we no longer obtain the exact solution and yet it is necessary to calculate the rates $\lambda(j)$ from the vector p--which is precisely what we are trying to avoid.

We shall describe our second approximation, first in terms of the reference example, and then in more general terms. For the reference example, since there are two stations, we shall consider two different aggregation steps. The first is that already used in section 3.3:

viz: $S = \bigcup S_1(i,k)$

The second is of the same type, but this time is defined with respect to station 2:

viz: $S = \bigcup S_2(i,k)$

where

$$S_2(i,k) = \{s \mid n_2=j, \ell_2=k\}$$

We therefore obtain two transition rate matrices relative to the two different aggregations i.e. $Q(1) \equiv Q$ and $Q(2)$.

Let $w(1)$ and $w(2)$ be the normalized solutions of

$w(1)^t Q(1)=0$ and $w(2)^t Q(2)=0$ respectively.

Now define $\nu_i(n_i)$ to be the departure rate of a customer from station i, knowing that there are n_i customers in station i. This rate may be expressed uniquely as a function of the vector w(i) and the parameters of the network. For example, for i=1

$$\nu_1(j) = \frac{\sum\limits_{\{k \mid n_1(k)=j\}} w_k(1) \quad \mu_{1\ell_1(k)} (1-a_{1\ell_1(k)})}{\sum\limits_{\{k \mid n_1(k)=j\}} w_k(1)} \tag{4}$$

Numerically, for the example considered, we have, (since we know the vector w(1))

$$\nu_1(0) = 0$$

$$\nu_1(1) = \frac{\mu_1\omega_3}{\omega_2+\omega_3} = \frac{\mu_1 \ \phi(\phi+3)}{2(\phi(\phi+3)+1)}$$

$$\nu_1(2) = \frac{\mu_1 \ \omega_5}{\omega_4+\omega_5} = \frac{\mu_1(3\phi+1)}{2(2\phi+1)}$$

In terms of the reference example, the approximation consists in taking approximate matrices $\hat{Q}(i)$ $i=1,2$ by using

$$\hat{\lambda}_i(j,k) = \hat{\nu}_{3-i}(N-j) \quad i=1,2$$

where the $\hat{\nu}_i(j)$ are computed by means of the vectors $\hat{w}(i)$ which are the solutions of $\hat{w}^t(i)\hat{Q}(i)=0$. Since $\hat{\lambda}_i(j,k) = \hat{\lambda}_i(j,k')$, $\forall\, k', k$, we may use the notation $\hat{\lambda}(j)$. The entire set is solved iteratively by taking $\nu_i^{(0)}(j)$ as an initial estimation. (for example, $\nu_i^{(0)}(j) = \frac{1}{m_i}$, $j=1,2$ where m_i is the mean service time for station i, $i=1,2$).

This then gives us an initial pair of matrices $\hat{Q}(1)$ and $\hat{Q}(2)$. From these matrices we may compute the vectors $\hat{w}(1)$ and $\hat{w}(2)$ which allow us to obtain new values for $\hat{\nu}_i(j)$ and so on. It is important to note that no approximation to the vector p is ever used.

Note that even if we start with the exact $\lambda_i(j)$'s, we obtain only an approximate vector $\hat{w}(i)$, $i=1,2$ since they are derived from the approximate matrices $\hat{Q}(i)$'s. Therefore from (4) we get approximate values $\hat{\nu}_i(j)$, $i=1,2$ which give approximate values $\hat{\lambda}_i(j)$ for the $\hat{\lambda}_i(j)$'s of the second iteration.

Note also that for this reference example because M=2, we have no difficulty in expressing the $\hat{\lambda}_i(j)$'s as functions of the $\hat{\nu}_i(j)$'s since for the exact values, we have exactly

$$\lambda_i(j) = \nu_{3-i}(N-j) \quad i=1,2$$

However, when M is greater than 2, this is no longer true and the exact values of the $\lambda_i(j)$'s cannot be determined from the $\nu_i(j)$'s without knowing some components of the vector p.

Consequently, to move from the reference example to the case of a general network, we also have to guess a good approximation to the $\lambda_i(j)$'s which depends only on the $\nu_i(j)$'s (in order not to use the vector p).

We consider M aggregations relative to the M stations. If $\hat{\nu}$ denotes the vector of throughput rates of the stations, we consider the product form network R* which has the same topology as R and for which the service rates are given by the vector $\hat{\nu}$. We compute the conditional arrival rates $\lambda_i^*(j)$, $i=1,2,\ldots M$, $j=0,\ldots N-1$ by means of the usual product form algorithms. Then we consider successively, the M phases of aggregation by taking $\hat{\lambda}_i(n_i,\ell_i) = \hat{\lambda}_i(n_i) = \lambda_i^*(n_i)$ which allows us to obtain a new vector $\hat{\nu}$. This last approximation corresponds exactly to the iterative method presented in section 3.2.

Thus, by considering M phases of aggregation, and no disaggregation step we obtain an approximate method which does not use the potentially very large stationary probability vector p.

4. Conclusions

In this paper we have examined the iterative method of Takahashi as applied to general queueing networks. It was shown that, as far as the aggregation step is concerned, the only interblock transition rates that could not be computed exactly a priori, where those which corresponded to the arrival of a customer to a designated station. Furthermore, it was observed that this aggregation step corresponds to the solution of the queue $\lambda(n,\ell)|C_{L_1}|1|N$, i.e. a queue for which the service time distribution is represented by a law of Cox of order L_1 and for which the arrival process is a stochastic point process whose intensity $\lambda(n,\ell)$ depends on the state of the system. The desirability of including a disaggregation step which involves the enumeration of all the states of the underlying Markov process was questioned. The iterative technique of Marie was seen to consist of many aggregation steps, one corresponding to each station of the network, but no disaggregation step. It is hoped that this work may lead to a better understanding and perhaps even a quantification of the errors involved in the iterative method of Marie.

References

BALB 79 Balbo, G., "Approximate Solutions of Queueing Network Models of Computer Systems", Ph.D. Thesis, Dept. of Computer Science, Purdue University, December, 1979.

BASK 75 Baskett, F., K. M. Chandy, R. R. Muntz and J. Palacios, "Open, Closed and Mixed Networks with Different Classes of Customers", J. ACM 22, 2, April, 1975, pp. 248-260.

CHAN 75 Chandy, K. M., U. Herzog and L. Woo, "Approximate Analysis of General Queueing Networks", IBM J. Res. Dev., 19, 1, January, 1975, pp. 43-49.

COUR 77 Courtois, P. J., Decomposability: Queueing and Computer Systems Applications, Academic Press, New York, 1977.

KOUR 83 Koury, J. R., D. F. McAllister and W. J. Stewart, "Iterative Methods for Computing Stationary Distributions of Nearly Completely Decomposable Markov Chains", to appear, SIAM Journal on Algebraic and Discrete Methods, 1983.

MARI 78 Marie, R. A., "Modelisation par Reseaux de Files d'Attente", Thesis: Docteur-es-Sciences Mathematiques, Université de Rennes, France, November, 1978.

MARI 83 Marie, R. A., and J. Pellaumail, "Steady State Probabilities for a Queue with a General Service Distribution and State Dependent Arrivals", IEEE Transactions on Software Engineering, Vol. SE-9, No. 1, January, 1983, pp. 109-113.

MCAL 83 McAllister, D. F., G. W. Stewart and W. J. Stewart, "On a Raleigh-Ritz Refinement Technique for Nearly Uncoupled Stochastic Matrices", to appear in Linear Algebra and Its Applications, 1983.

STEW 80 Stewart, G. W., "Computable Error Bounds for Aggregated Markov Chains", Technical Report 901, Computer Science Dept., University of Maryland, May, 1980.

STEW 78 Stewart, W. J., "A Comparison of Numerical Techniques in Markov Modelling", CACM, Vol. 21, February, 1978, pp. 144-151.

TAKA 75 Takahashi, Y., "A Lumping Method for Numerical Calculations of Stationary Distributions of Markov Chains", Dept. of Information Sciences, Tokyo Institute of Technology, Japan, Research Report B-18, June, 1975.

VANT 81 Vantilburgh, H., "The Error of Aggregation. A Contribution to the Theory of Decomposable Systems and Applications", Thesis: Docteur-es-Sciences Appliquies, Université Catholique de Louvain, Belgium, 1981.

Liste des Publications Internes IRISA

PI 172 **Comparaison de groupes de variables : 2ème partie : un exemple d'application**
B. Escofier, J. Pajès, 37 pages ; Juillet 1982

PI 173 **Unfold-fold program transformations**
L. Kott, 29pages ; Juillet 1982

PI 174 **Remarques sur les langages de parenthèses**
J.M. Autebert, J. Beauquier, L. Boasson, G. Senizergues, 20 pages ; Juillet 1982

PI 175 **Langages de parenthèses, langages N.T.S. et homomorphismes inverses**
J.M. Autebert, L. Boasson, G. Senizergues, 26 pages ; Juillet 1982

PI 176 **Tris pour machines synchrones ou Baudet Stevenson revisited**
R. Rannou, 26 pages ; Juillet 1982

PI 177 **Un nouvel algorithme de classification hiérarchique des éléments constitutifs de tableau de contingence basé sur la corrélation**
B. Tallur, Juillet 1982 ;

PI 178 **Programmes d'analyse des re'sultats d'une classification automatique**
I.C. Lerman et collaborateurs, 79 pages ; Septembre 1982

PI 179 **Attitude à l'égard des mathématiques des élèves de sixième**
J.Degouys, R. Gras, M. Postic, 29 pages ; Septembre 1982

PI 180 **Traitements de textes et manipulations de documents : bibliographie analytique**
J. André, 20 pages ; Septembre 1982

PI 181 **Algorithme assurant l'insertion dynamique d'un processeur autour d'un réseau à diffusion et garantissant la cohérence d'un système de numérotation des paquets global et réparti**
Annick Le Coz, Hervé Le Goff, Michel Ollivier, 31 pages ; Octobre 1982

PI 182 **Interprétation non linéaire d'un coefficient d'association entre modalités d'une juxtaposition de tables de contingence**
Israël César Lerman, 34 pages ; Novembre 1982

PI 183 **L'IRISA vu à travers les stages effectués par ses étudiants de DEA (1è année de thèse)**
Daniel Herman, 41 pages ; Novembre 1982

PI 184 **Commande non linéaire robuste des robots manipulateurs**
Claude Samson, 52 pages ; Janvier 1983

PI 185 **Dialogue et représentation des informations dans un système de messagerie intelligent**
Philippe Besnard, René Quiniou, Patrice Quinton, Patrick Saint-Dizier, Jacques Siroux, Laurent Trilling, 45 pages ; Janvier 1983

PI 186 **Analyse classificatoire d'une correspondance multiple ; typologie et régression**
I.C. Lerman, 54 pages ; Janvier 1983

PI 187 **Estimation de mouvement dans une sequence d'images de télévision en vue d'un codage avec compensation de mouvement**
Claude Labit, 132 pages ; Janvier 1983

PI 188 **Conception et réalisation d'un logiciel de saisie et restitution de cartes élémentaires**
Eric Sécher, 45 pages ; Janvier 1983

PI 189 **Etude comparative d'algorithmes pour l'amélioration de dessins au trait sur surfaces point par point**
M.A. ROY, 96 pages ; Janvier 1983

PI 190 **Généralisation de l'analyse des correspondances à la comparaison de tableaux de fréquence**
Brigitte Escofier, 35 pages ; Mars 1983

PI 191 **Association entre variables qualitatives ordinales «nettes» ou «floues»**
Israël-César Lerman, 42 pages ; Mars 1983

PI 192 **Un processeur intégré pour la reconnaissance de la parole**
Patrice Frison, 80 pages ; Mars 1983

PI 193 **The Systematic Design of Systolic Arrays**
Patrice Quinton, 39 pages ; Avril 1983

PI 194 **Régime stationnaire pour une file M/H/1 avec impatience**
Raymond Marie et Jean Pellaumail, 8 pages ; Mars 1983

PI 195 **SIGNAL : un langage pour le traitement du signal**
Paul Le Guernic, Albert Benveniste, Thierry Gautier, 49 pages ; Mars 1983

PI 196 **Algorithmes systoliques : de la théorie à la pratique**
Françoise André, Patrice Frison, Patrice Quinton, 19 pages ; Mars 1983

PI 197 **HAVANE : un système de mise en relation automatique de petites annonces**
Patrick Bosc, Michèle Courant, Sophie Robin, Laurent Trilling, 79 pages ; Mai 1983

PI 198 **Une procédure de décision en logique non-monotone**
Philippe Besnard, 59 pages ; Mai 1983

PI 199 **A formal proof system for infinitary rational expressions**
Philippe Darondeau, Laurent Kott, 28 pages ; Mai 1983

PI 200 **Etude générale d'un réseau constitué de deux stations hyperexponentielles**
Jean-Yves Le Boudec, 12 pages ; Mai 1983

PI 201 **Langage de Dyck et groupe symétrique**
Yves Cochet, 13 pages ; Juin 1983

PI 202 **On the observational semantics of pair parallelism**
Philippe Darondeau, Laurent Kott, 14 pages ; Juin 1983

PI 203 **Les langages fonctionnels : caractéristiques, utilisation et mise en œuvre**
Daniel Le Métayer, 162 pages ; Juin 1983

PI 204 **On exact and approximate itterative methods for general queueing networks**
Raymond A. Marie, William J. Stewart, 27 pages ; Juin 1983